Solving the Secular Equation Including Spin Orbit Coupling for Systems with Inversion and Time Reversal Symmetry

J. J. DONGARRA,* J. R. GABRIEL,* D. D. KOELLING,† AND J. H. WILKINSON‡

Mathematics and Computer Science Division,
Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439

A computational method for computing the eigenvalues and eigenvectors of a class of matrices that arise in quantum mechanics involving time reversal and inversion symmetry is described. The algorithms presented have greatly reduced the computational effort required to solve this problem and also produce a stable, more accurate solution.

1. INTRODUCTION

Molecules and solids containing heavy atoms require the use of relativistic kinematics in the calculation of their electronic structure, because of the large velocities acquired by the electrons in the deep potentials near the nuclei. Relativistic kinematics introduces effects of two types (1) those that do not change the symmetry of the problem, such as the mass–velocity and Darwin terms, and (2) those that do modify symmetry, such as the spin–orbit coupling. Because the kinematic effects of the first type can easily be included at little or no additional computational expense and leave the symmetry analysis on a familiar basis, considerable effort has been expended to excise the spin–orbit coupling from the Dirac equation [1–5]. Nevertheless, it is important—especially in the case of the fifth series and the actinides—to include the spin–orbit effects. Thus, the more complicated problem must be dealt with, at greatly increased effort (and cost).

The inclusion of spin–orbit coupling into a calculation is not difficult in principle, but it does double the size of the matrices involved in the calculation and require that the matrices contain complex elements. This has the potential for increasing computation times by between one and two orders of magnitude. Such an increase need not occur, however, if the system has both time reversal (i.e., no magnetic fields or moments) and parity (spatial inversion) symmetry. In that case, all eigenvalues are

278

doubly degenerate. Soven [6] has described a technique to directly evaluate the term that, when squared, is the determinant of the secular equation. This scheme has been used extensively in relativistic augmented plane wave (RAPW) calculations [7] which search for eigenvalues by seeking zeroes of the determinant. Soven's scheme speeds up the calculation to the extent that the cost of including spin–orbit coupling is no more than a factor of two over comparable non-relativistic schemes when applied to procedures where the nonlinear energy variation is done by searching for the zeroes of the determinant of the secular equation. More recently, RAPW schemes have been linearized [8–10] such that it is more appropriate to solve a generalized eigenvalue problem than to plot determinants. In fact, even for the standard nonlinear APW techniques, it is often better to solve a generalized eigenvalue problem coupled with a parameter variation than to plot determinants [11].

Techniques have now been developed and are presented here to make the generalized eigenvalue problem more efficient. These techniques are similar to methods that the Soven scheme yields for determinants. It should be pointed out that there are several equally compelling reasons for incorporating such techniques besides the increased speed that is achieved. Most easily perceived is that the memory requirements for the problem are reduced by half. Furthermore, for an RAPW basis set, the accuracy is greatly enhanced. To understand this latter point, one must remember that the RAPW basis set is, in principle, overcomplete. (A full plane wave basis—which is complete over all space—labels that part of the basis functions applied to the incomplete space of the interstitial region.) In practice, the necessity of truncating the expansion removes the problem. For a reasonably converged calculation without spin–orbit, the eigenvalue spectrum of the overlap matrix ranges from 1 down to 10^{-5}, with one eigenvalue split off at 10^{-9}. With care, one can still solve the generalized problem by performing a Cholesky decomposition and diagonalization. When spin orbit coupling is included, however, several things happen. The matrix is doubled in size and becomes complex. Thus the numerical noise is dramatically increased because of the significant increase in operations to be performed. In addition, all eigenvalues are doubled, so that the smallest eigenvalue occurs twice. The result is that one cannot directly perform a Cholesky decomposition on the expanded matrix for an adequate basis set (ε^2 is, after all, much smaller than ε as $\varepsilon \rightarrow 0$!). This has made it necessary to introduce schemes that, although very effective for calculations on elemental systems, are quite expensive for compound systems.

The essential feature of the generalized secular equation with spin–orbit included is that it can be written in terms of elements that are real multiples of the $SU(2)$ matrices (i.e., quaternions) when one has the required time reversal and parity symmetry. This is easily seen for the various forms of RAPW methods, since the Hamiltonian can be written as

$$H(\mathbf{k}_1 s_1 ; \mathbf{k}_2 s_2) = H_0(\mathbf{k}_1 \mathbf{k}_2) \, \delta_{s_1 s_2} + H_{s0}(\mathbf{k}_1 s_1 ; \mathbf{k}_2 s_2), \tag{1.1}$$

$$H_{s0}(\mathbf{l}_1 s_1 ; \mathbf{k}_2 s_2) = ih(|\mathbf{k}_1|, |\mathbf{k}_2|, \hat{k}_1 \cdot \hat{k}_2)(\hat{k}_1 \times \hat{k}_2) \cdot \sigma_{s_1 s_2}, \tag{1.2}$$

where $(\mathbf{k}, s)$ labels the plane wave part of the basis function in the interstitial region; σ are the Pauli spin matrices; H_0 is the nonrelativistic matrix with the potential parameters replaced by properly averaged relativistic ones; and H_0 and h are real. The overlap matrix S will have an identical structure. We emphasize the structure, because it is the crucial feature for the techniques to follow and the precise form of H_0 and h will depend on the details of the augmentation chosen. The spin index in Eq. (1.1) does not imply that the basis functions are of a definite spin. They are taken to have a definite spin in the interstitial region where the spin–orbit coupling is negligible, enabling us to use a definite spin index there. However, as the nucleus is approached, the spin–orbit coupling acts and the spin is mixed. For H_0 and h to be real, it is necessary to choose the origin at an inversion site. Fortunately, such a choice introduces many additional computational efficiences.

From Eqs. (1.1) and (1.2), one sees that the effect of including spin–orbit coupling is to replace each scalar matrix element in the nonrelativistic or spin–orbitless problem with a complex 2×2 matrix of the form

$$\alpha \equiv \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}. \tag{1.3}$$

It is quite interesting to view this as a case of replacing the real scalar matrix elements in the spin–orbitless case by a set of hypercomplex numbers or quaternions. These numbers add and multiply since the resultant will also have the structure of (1.3). Further, they have an easy division since

$$\alpha^+ \alpha = (|a|^2 + |b|^2) I_2 \equiv r^2 I_2. \tag{1.4}$$

(Note that α^+ is the complex conjugate transpose of α and I_2 is a 2×2 identity matrix.) They do not commute, but commutivity is not required (except by the diagonal elements which do commute) to calculate a determinant, perform a Cholesky decomposition, or diagonalize a matrix. Thus one can perform any of these three operations using the quaternion form. This, then, can be taken as the formal basis for the Soven technique. Note that it also leads to a very natural demonstration of the double degeneracy (also using the Hermitian properties, of course) since a real diagonal matrix of quaternions would contain each scalar eigenvalue twice.

The inclusion of spin–orbit coupling will generate the quaternion form as well as methods utilizing other basis sets. This can be shown by properly accounting for the "moment-flip" operator

$$Q = JK \tag{1.5}$$

where J is the parity operator and K the time reversal operator. Normally one represents

$$J = \beta I, \tag{}$$

$$\beta = \begin{pmatrix} I_2 & 0 \\ 0 & I_2 \end{pmatrix}, \tag{1.6}$$

and

$$K = i\sigma_y C, \qquad (1.7)$$

where I is the spatial inversion operator and C is the complex conjugation operator. We require that Q commutes with the Hamiltonian. Q is a very unusual operator in that $Q^2 = -1$ and $\langle x| Q^+ |y\rangle = -\langle x| Q |y\rangle^*$. (It is antiunitary.) Given the basis function $|y\rangle$, the basis function $Q|y\rangle$ will be orthogonal to it,

$$\langle y| Q |y\rangle = \langle y| Q^+ |y\rangle^* = -\langle y| Q |y\rangle,$$

using first the Hermitian property of the scalar product and then the Q^+ relation. Hence, it is reasonable to pair $|y\rangle$ and $Q|y\rangle$. The 2×2 matrix between the basis pairs $|x\rangle$, $Q|x\rangle$ and $|y\rangle$, $Q|y\rangle$ will have the quaternion form,

$$\langle x| Q^+ Q |y\rangle = -\langle x| Q^2 |y\rangle^* = \langle x | y\rangle^*,$$

$$\langle x| Q^+ |y\rangle = -\langle x| Q |y\rangle^*,$$

so

$$\begin{pmatrix} \langle x| y\rangle & \langle x| Q |y\rangle \\ \langle x| Q^+ |y\rangle & \langle x| Q^+ Q |y\rangle \end{pmatrix} = \begin{pmatrix} \langle x| y\rangle & \langle x| Q |y\rangle \\ -\langle x| Q |y\rangle^* & \langle x| y\rangle^* \end{pmatrix}.$$

Clearly the Hamiltonian matrix will have the same properties as the overlap matrix displayed, since $[Q, H] = 0$. In the case of a plane wave representation (APW or OPW), Q acts as a spin–flip operator, and the basis function pairs are the two spin states. In the case of an angular momentum representation (KKR, LMTO, ASW, or LCAO), Q translates (j, μ) to $(j, -\mu)$ on the site related by inversion symmetry. These are the magnetic moment-related states that have been the basis for a generalization of spin polarized calculations in the presence of spin–orbit coupling [12]. Here, of course, we use them merely to obtain a quaternion format.

2. COMPUTATIONAL TECHNIQUES

One can generalize the standard techniques such as described in Wilkinson [13] and Stewart [14] in terms of quaternions; i.e., given a quaternion data type one can generalize the algorithms to operate on quaternion elements instead of complex or real elements. The increased complexity of the quaternion arithmetic can involve more computational effort than merely doubling the matrix size and dealing with scalars. However, one achieves economy of storage and increased accuracy. In the case of determinant calculation, the rewriting of the quaternion arithmetic into scalar operations is the Soven technique. In the case of the generalized eigenvalue problem, it is only part of the solution. By a careful organization of the calculations, it is possible to arrange the majority of operations involving quaternions to multiplications involving real scalar multiples of the identity matrix [15]. This

DONGARRA ET AL.

reduces the computational effort for that part of the reduction to the level of the spin–orbitless problem. To facilitate writting the necessary programs, one can implement a quaternion abstract data-type and then construct the algorithms directly in the abstract language. Application programs can now be written using a quaternion data type and be translated to Fortran automatically [16, 17]. We have choosen to implement the algorithms described here using conventional Fortran directly, this was done mainly to fully minimize the space and time required to solve the problem.

In this section, we will expand the quaternion arithmetic to clarify the reorganization actually used in the computer codes realizing the technique. We will first examine the standard eigenvalue problem

$$Hx = \lambda x. \tag{2.1}$$

In terms of quaternions, the matrix H has the form

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{12}^+ & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{13}^+ & \alpha_{23}^+ & \alpha_{33} & \alpha_{34} \\ \alpha_{14}^+ & \alpha_{24}^+ & \alpha_{34}^+ & \alpha_{44} \end{pmatrix}. \tag{2.2}$$

Expanding these elements in terms of complex elements, the matrix has the form

a_{11}	0	a_{12}	b_{12}	a_{13}	b_{13}	a_{14}	b_{14}
0	a_{11}	$-b_{12}^*$	a_{12}^*	$-b_{13}^*$	a_{13}^*	$-b_{14}^*$	a_{14}^*
a_{12}^*	$-b_{12}$	a_{22}	0	a_{23}	b_{23}	a_{24}	b_{24}
b_{12}^*	a_{12}	0	a_{22}	$-b_{23}^*$	a_{23}^*	$-b_{24}^*$	a_{24}^*
a_{13}^*	$-b_{13}$	a_{23}^*	$-b_{23}$	a_{33}	0	a_{34}	b_{34}
b_{13}^*	a_{13}	b_{23}^*	a_{23}	0	a_{33}	$-b_{34}^*$	a_{34}^*
a_{14}^*	$-b_{14}$	a_{24}^*	$-b_{24}$	a_{34}^*	$-b_{34}$	a_{44}	0
b_{14}^*	a_{14}	b_{24}^*	a_{24}	b_{34}^*	a_{34}	0	a_{44}

where the a_{ij} and b_{ij} are complex and the a_{ii} are real.

We will show that the matrix can be reduced to the form

$$\begin{pmatrix} T & 0 \\ 0 & T \end{pmatrix}, \tag{2.3}$$

where T is a real symmetric $n \times n$ tridiagonal matrix. There are $n - 2$ major steps each of which consists of two minor steps. These steps have a great deal in common with those for the real symmetric eigenvalue problem.

In the first minor step, we use Eq. (1.4) to construct transformations that change a column and row of quaternion elements into real elements. These transformations have the form

$$\frac{a}{r},\tag{2.4}$$

which is a unitary 2×2 matrix. As the first stage of reducing a row and column the matrix to tridiagonal form, we perform a unitary similarity transformation on the matrix represented in Eq. (2.2) with the 2×2 block diagonal matrix defined by

$$\begin{bmatrix} I_2 & & & \\ & \dfrac{a_{12}}{r_{12}} & & \\ & & \dfrac{a_{13}}{r_{13}} & \\ & & & \dfrac{a_{14}}{r_{14}} \end{bmatrix}.\tag{2.5}$$

After this has been applied, the transformed matrix is of the form

$$\left[\begin{array}{cc|cc|cc|cc}
a_{11} & 0 & r_{12} & 0 & r_{13} & 0 & r_{14} & 0 \\
0 & a_{11} & 0 & r_{12} & 0 & r_{13} & 0 & r_{14} \\
\hline
r_{12} & 0 & a_{22} & 0 & a_{23} & b_{23} & a_{24} & b_{24} \\
0 & r_{12} & 0 & a_{22} & -b_{23}^* & a_{23}^* & -b_{24}^* & a_{24}^* \\
\hline
r_{13} & 0 & a_{23}^* & -b_{23} & a_{33} & 0 & a_{34} & b_{34} \\
0 & r_{13} & b_{23}^* & a_{23} & 0 & a_{33} & -b_{34}^* & a_{34}^* \\
\hline
r_{14} & 0 & a_{24}^* & -b_{24} & a_{34}^* & -b_{34} & a_{44} & 0 \\
0 & r_{14} & b_{24}^* & a_{24} & b_{34}^* & a_{34} & 0 & a_{44}
\end{array}\right],\tag{2.6}$$

where a_{ij}, b_{ij} now denote values after the transformations. These new values are derived via relations of the type exemplified by

$$\begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}\begin{pmatrix} x & y \\ -y^* & x^* \end{pmatrix} = \begin{pmatrix} ax - by^* & bx^* + ay \\ -b^*x - a^*y^* & a^*x^* - b^*y \end{pmatrix} = \begin{pmatrix} u & v \\ -v^* & u^* \end{pmatrix},$$

where

$$u = ax - by^*, \qquad v = bx^* + ay.$$

so that the product of matrices of the form

$$\begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix}$$

is a matrix of the same form. This is the multiplicative property of the quaternion elements mentioned in the previous section. Matrices of this type generally do not commute, although they do so if all of the elements are real. Fortunately, we do not need commutivity except on the diagonal where the elements are real. Notice now that the first two rows and columns of the matrix are real.

We have now completed the first minor step of the first major step. In the second minor step, we perform the quaternion analog of the classical Householder tridiagonalization. This will eliminate all the quaternion blocks beyond the (quaternion) subdiagonal. A very important feature here is that the quaternion elements of the transformation are now real multiples of the 2×2 unit matrix. Thus their arithmetic is no more involved than that for real scalars. This second minor step completes the major step. The transformed matrix is of the form

$$\left(\begin{array}{cc|cc|cc|cc} a_{11} & 0 & r_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{11} & 0 & r_{12} & 0 & 0 & 0 & 0 \\ \hline r_{12} & 0 & a_{22} & 0 & a_{23} & b_{23} & a_{24} & b_{24} \\ 0 & r_{12} & 0 & a_{22} & -b_{23}^* & a_{23}^* & -b_{24}^* & a_{24}^* \\ \hline 0 & 0 & a_{23}^* & -b_{23} & a_{33} & 0 & a_{34} & b_{34} \\ 0 & 0 & b_{23}^* & a_{23} & 0 & a_{33} & -b_{34}^* & a_{34}^* \\ \hline 0 & 0 & a_{24}^* & -b_{24} & a_{34}^* & -b_{34} & a_{44} & 0 \\ 0 & 0 & b_{24}^* & a_{24} & b_{34}^* & a_{34} & 0 & a_{44} \end{array} \right) \tag{2.7}$$

which has the same structure as the original matrix except for the first two rows and columns, which have been processed. After the completion of $n - 2$ major steps and the first part of the $(n - 1)$th step, the matrix has the form

$$\left(\begin{array}{cc|cc|cc|cc} a_{11} & 0 & r_{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{11} & 0 & r_{12} & 0 & 0 & 0 & 0 \\ \hline r_{12} & 0 & a_{22} & 0 & r_{23} & 0 & 0 & 0 \\ 0 & r_{12} & 0 & a_{22} & 0 & r_{23} & 0 & 0 \\ \hline 0 & 0 & r_{23} & 0 & a_{33} & 0 & r_{34} & 0 \\ 0 & 0 & 0 & r_{23} & 0 & a_{33} & 0 & r_{34} \\ \hline 0 & 0 & 0 & 0 & r_{34} & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{34} & 0 & a_{44} \end{array} \right) \tag{2.8}$$

At this point the matrix can be separated easily into two identical tridiagonal matrices of the form represented in (2.3) by a simple rearrangement of the rows and columns. The eigenvalues and eigenvectors of the tridiagonal matrix can be determined easily by standard techniques [16]. The eigenvectors of the original problem can be found by applying, in the appropriate way, the transformations used in the reduction to the eigenvectors of the tridiagonal matrix.

3. THE GENERALIZED PROBLEM

The standard problem as expressed in Eq. (2.1) has been presented first for convenience, but in practice the problem commonly arises in the form

$$Hz = \lambda Sz, \tag{3.1}$$

where both H and S have the same structure as H in the previous section and S is positive definite. The generalized problem can be reduced to the standard problem if we can determine the matrix U such that

$$USU^H = I. \tag{3.2}$$

We have then

$$UHU^H(U^{-H}z) = \lambda USU^H(U^{-H}z). \tag{3.3}$$

Obviously, for economy of computation, it is desirable to determine U in a factorized form

$$U = U_k \cdots U_2 U_1 \tag{3.4}$$

and in such a way that the

$$U_1 HU_1^H \qquad\qquad U_1 SU_1^H$$
$$U_2 U_1 HU_1^H U_2^H \qquad U_2 U_1 SU_1^H U_2^H$$
$$\vdots \qquad\qquad\qquad \vdots$$
$$UHU^H \qquad\qquad\qquad I$$

have the same structure as H at every stage.

To do so requires $(n-1)$ major steps, each step being determined by the current S matrix. The first major step is wholly typical. It again consists of two minor steps. The first minor step is exactly the same as that applied to H in the previous section. This reduces S to the form illustrated in (2.6)—the row and column being worked on are reduced to real multiples of the 2×2 unit matrix.

In the second minor step, we will annihilate the off-diagonal blocks in the rows and columns reduced during this step. Since S is positive definite, we can carry out this

reduction by performing one phase of a Cholesky factorization at each major step on the quaternion form. The structure of S is obviously preserved. If we think in terms of the real and the imaginary parts of S, this second minor step obviously affects only the real part.

After $(n-1)$ steps of this kind, S is reduced to a real positive diagonal matrix. This can be reduced to the identity matrix by pre-multiplication and post-multiplication by the reciprocal square root of the diagonal matrix. All transformations applied to S must also be applied to H. The structure of H is obviously preserved, although, of course, it remains a full matrix (i.e., no zeroes are induced). More work is involved in the transformations applied to H than of S both for this reason and also because all off-diagonal elements of H remain complex throughout. Once S has been reduced to the identity matrix the process described in Section 2 can be used to reduce H to diagonal form.

4. IMPLEMENTATION

For the software implementation, only part of the information represented in Eq. (3.1) need actually be stored, and many of the arithmetic operations are not performed because of the obvious symmetries in the processes.

Each of the matrices, H or S, can be represented mathematically by a quaternion matrix of order n. Each of the elements in the quaternion matrix has the form of Equation (1.3), and can be defined by the two quanties a and b. Which require two complex numbers or four real elements for their representation. Since the matrices H and S are Hermitian, only half the elements are necessary. Thus, for each matrix of n^2 quaternion elements, $2n^2$ real locations are required. In the implementation, we have chosen to segregate the elements so one $n \times n$ array holds elements corresponding to the a's and another $n \times n$ array holds elements corresponding to the b's. The lower triangular part of each array contains the real part of the elements and the upper triangular part the imaginary part. Thus, in order to determine the i, j element of the quaternion matrix, references are made to the i, j and j, i elements of each array in storage.

The algorithm used in the implementation is presented here to illustrate the structure and give a feeling for the operational complexity of the procedure. We start with the reduction of the matrix S in Eq. (3.1) to diagonal form.

Reduction of S to Diagonal Form

> For $j = 1$ to $n - 1$
> > For $i = j + 1$ to n
> > > Call *giveng*(ij, Sa, Sb, Ha, Hb)
> > End
> > Call *cholsky*(j, Sa, Sb, Ha, Hb)
> End
> *Apply Diagonal Matrix to Ha and Hb*

Routines *giveng* and *cholsky* used above apply their transformations to H. The next step is to reduce H to tridiagonal form and determine the eigenvalues and eigenvectors of the reduced system.

Determine the Eigenvalues of H

> For $j = 1$ to $n - 2$
> > For $i = j + 1$ to n
> > > Call *given*(i, j, Ha, Hb)
> >
> > End
> > Call *house*(j, Ha, Hb)
>
> End
> Call *given*($n, n - 1, Ha, Hb$)
> *Find Eigenvalues and Eigenvectors of Tridiagonal Matrix*

The routines *given* and *giveng* apply transformations of the form illustrated in Eq. (2.4). The loops on i in the above procedures correspond to application of the matrix described in Eq. (2.5). The values of i and j reflect the rows and columns that are effected by the transformation. The two routines differ only in that *giveng* applies the transformations to H as well as to S.

The arithmetic performed by *given* and *giveng* will be totally complex. They introduce a zero and produce a real element into the matrix. The routines *cholsky* and *house*, in a sense, remove this real quantity produced by *given* and by *giveng*. They perform real arithmetic on the complex matrix entries. In terms of operation counts, when *given* is invoked there are $n - j + 1$ elements involved in the process and each element is involved in 4 complex multiplies operations. The same is true for *giveng* plus $4n$ complex multiplies as a result of applying the transformations to H.

5. APPLICATION

No detailed evaluation of the timings of this approach is available. We can, however, offer some observations that have been made in the course of applying it in a number of applications. Rough timings indicate that the inclusion of spin–orbit effects in both the Hamiltonian H and the overlap matrix S requires 5–6 times the computational effort rather than the factor of 16 one might have experienced otherwise. Because not as much calculation can be avoided in the Cholesky transformation, the situation is much better when the overlap matrix need not be considered in quaternion format. This occurs when the spin–orbit coupling need not modify the basis functions so they can be treated as product functions of spin and coordinate space. The overlap matrix is then spin diagonal. This is easily achieved for the lighter elements up through the rare earths but is not really an acceptable approximation through the $5d$'s and into the actinides. The diagonalization alone is six times faster than a straightforward doubling of the matrix size making the efficiency gained roughly comparable to that achieved in the evaluation of determinants.

Previous techniques for dealing with the problem of the spin orbit coupled matrices in the APW formalism have involved solving the problem with the spin–orbit coupling excised and using the lower energy subset of eigenvectors to perform a second restricted variation including the spin–orbit coupling. In the material $CeRh_3$, errors as large as four milliRydberg were found in the results for this approximate method with the truncation set that was used. By using the quaternion formulation, one achieves roughly the same computational efficiency but with better accuracy.

REFERENCES

1. D. D. KOELLING AND B. N. HARMON, *J. Phys. C* **10** (1977), 3107.
2. J. H. WOOD AND A. M. BORING, *Phys. Rev. B* **18** (1978), 2701.
3. H. GOLLISCH AND L. FRITSCHE, *Phys. Status Solidi B* **86** (1978), 145.
4. F. ROSICKY AND F. MARK, *J. Phys. B* **8** (1975), 2581; F. ROSICKY, P. WEINBERGER, AND F. MARK, *J. Phys. B* **9** (1976), 2971.
5. T. TAKEDA, *Z. Phys. B* **32** (1978), 43.
6. P. SOVEN, *Phys. Rev.* **137** (1965), A1706.
7. T. L. LOUCKS, "Augmented Plane Wave Method: A Guide to Performing Electronic Structure Calculations," Benjamin, New York, 1967.
8. D. D. KOELLING, *in* "Proceedings, Second Intern. Conf. on the Electronic Struc. of the Actinides" (J. Mulak, W. Suski, and R. Troc, Eds.), Ossolineum, Wroclow, Poland, 1976.
9. A. H. MACDONALD, W. E. PICKETT, AND D. D. KOELLING, *J. Phys. C* **13** (1980), 2675.
10. T. TAKEDA AND J. KUEBLER, *J. Phys F* **9** (1979), 661; T. TAKEDA, *J. Phys. F* **9** (1979), 815.
11. B. N. HARMON AND D. D. KOELLING, *J. Phys C* **7** (1974), L210.
12. D. E. ELLIS AND G. L. GOODMAN, *Intl. J. Quant. Chem.*, in press.
13. J. H. WILKINSON, "The Algebraic Eigenvalues Problem," Oxford Univ. Press, London, 1965.
14. G. W. STEWART, "Introduction to Matrix Computation," Academic Press, New York, 1973.
15. J. J. DONGARRA, J. R. GABRIEL, D. D. KOELLING, AND J. H. WILKINSON, *Linear Algebra Appl.* **59** (1984), in press.
16. J. M. BOYLE, Program transformations and language design, *in* "The Relationalship Between Numerical Computations and Programming Languages," (J. K. Reid, Ed.), pp. 285–295, North-Holland, Amsterdam, 1982.
17. J. M. BOYLE, Private communications, 1982.
18. B. T. SMITH *et al.*, "Matrix Eigensystem Routines—EISPACK Guide," Lecture Notes in Computer Science, Vol. 6, 2nd ed., Springer-Verlag, Berlin, 1976.